# ATLAS
## MANUAL DE USUARIO
## COMPONENTE CODIGO DE BARRAS

Versión 1.3

Área de Aplicaciones Especiales y Arquitectura de Software




Agencia de Informática y Comunicaciones de la Comunidad de Madrid



**Hoja de Control**

| | | | |
|---|---|---|---|
| **Título** | Manual de Usuario Componente Código de barras | | |
| **Documento de Referencia** | NORMATIVA ATLAS | | |
| **Responsable** | Área de Aplicaciones Especiales y Arquitectura de Software | | |
| **Versión** | 1.3 | **Fecha Versión** | 19/07/2013 |

Registro de Cambios

| Versión | Causa del Cambio | Responsable del Cambio | Fecha del Cambio |
|---|---|---|---|
| 1.0 | Versión inicial del documento | Área de Integración y Arquitectura de Aplicaciones | 26/05/2010 |
| 1.1 | 2. Descripción: Actualizada captura del componente. | Área de Integración y Arquitectura de Aplicaciones | 15/02/2011 |
| 1.2 | Se modifica el nombre del Area | Área de Aplicaciones Especiales y Arquitectura de Software | 27/09/2011 |
| 1.3 | Modificación general del manual por la creación de un nuevo componente. | Área de Aplicaciones Especiales y Arquitectura de Software | 15/07/2013 |

## Índice


1. **INTRODUCCIÓN** .................... **4**
   - 1.1. AUDIENCIA OBJETIVO .................... 4
   - 1.2. CONOCIMIENTOS PREVIOS .................... 4
2. **DESCRIPCIÓN** .................... **4**
3. **INSTALACIÓN Y CONFIGURACIÓN** .................... **5**
4. **USO** .................... **6**
   - 4.1. PASO 1: DEFINICIÓN DEL ESPACIO DE NOMBRES DE ETIQUETAS DE ATLAS .................... 6
   - 4.2. PASO 2: INSERCIÓN EN LA PÁGINA DE LA ETIQUETA DE ATLAS .................... 6
   - 4.3. PRINCIPALES TIPOS DE CÓDIGOS .................... 8
   - 4.4. SERVICIO DE CÓDIGO DE BARRAS .................... 9
   - 4.5. RECOMENDACIONES Y BUENAS PRÁCTICAS .................... 10
   - 4.6. EJEMPLO DE USO .................... 10
5. **PREGUNTAS MÁS FRECUENTES** .................... **11**
6. **ENLACES RELACIONADOS** .................... **12**

# 1. INTRODUCCIÓN

Este documento contiene el manual de uso del componente visual "Código de barras" del framework Atlas. En él se incluye información sobre cómo utilizar dicho componente en una aplicación web, así como información acerca de la configuración de los parámetros fundamentales del componente.

## 1.1. AUDIENCIA OBJETIVO

Este documento está orientado a toda aquella persona que esté desarrollando una aplicación Web basada en el Framework Atlas y necesite utilizar componentes de presentación en su aplicación Web.

## 1.2. CONOCIMIENTOS PREVIOS

Para un completo entendimiento del documento, el lector deberá tener conocimientos previos sobre las siguientes tecnologías:

- Java Server Faces (JSF)
- Facelets
- Richfaces

Para saber más sobre dichas tecnologías, consultar el apartado de este documento, *Enlaces Relacionados*.

# 2. DESCRIPCIÓN

El componente Barcode es capaz de genera una imagen con un código de barras. Sus características principales son:

- Crea ficheros de imagen en varios formatos, conteniendo códigos de barras (1D) y puntos (2D).
- Permite mostrar códigos en aplicaciones Web.
- La generación del código se hace mediante ajax, utilizando componentes de la librería Richfaces.
- Soporta los siguientes Standard: Code128, Códigos QR, PDF 417, Code39, Aztec, Codabar, Datamatrix, Ean13, Ean8, ITF, UPC_A.
- Provee un servicio con métodos para generar los códigos en diferentes formatos y validar los parámetros de entrada.

Internamente, el componente utiliza la librería de google Zxing. Esta librería permite visualizar e imprimir diferentes formatos de código.

En la aplicación de componentes de Atlas hay un ejemplo en el que se puede probar el componente generando dinámicamente códigos de todos los tipos soportados modificando los parámetros para cada uno de ellos.

## 3. INSTALACIÓN Y CONFIGURACIÓN

El componente del código de barras ya viene instalado y configurado en el arquetipo Web, incluido con el módulo de componentes visuales. Por este motivo no es necesaria una instalación adicional si se parte del arquetipo.
En el caso de que se quiera hacer uso de los servicios que provee el componente, la clase AtlasFacade contiene los métodos necesarios, en el apartado 4.4 se detalla la configuración necesaria.

# 4. USO

Una vez instalado el módulo de componentes puede procederse a su utilización. Para ello deben realizarse los pasos indicados en los siguientes apartados:

## 4.1. Paso 1: Definición del espacio de nombres de etiquetas de Atlas

Es necesario crear un fichero *xhtml* y establecer la definición del espacio de nombres para las etiquetas de componentes de Atlas. Un ejemplo de cabecera de fichero *xhtml* es la siguiente:

**Ejemplo.xhtlm**

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
      "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="es"
      xmlns:ui="http://java.sun.com/jsf/facelets"
      xmlns:h="http://java.sun.com/jsf/html"
      xmlns:f="http://java.sun.com/jsf/core"
      xmlns:atlas="http://atlas.core.componentes/jsf">
```

## 4.2. Paso 2: Inserción en la página de la etiqueta de Atlas

Se incluirá una etiqueta atlas:barcode que es lo que define la imagen del código de barras en la página. A continuación se detalla dicha invocación.

**Ejemplo.xhtml**

```
<atlas:barcode id="codigodebarras" barcodeType="CODE_128"
            data="1234567890" height="70" width="200" />
```

Con el código anterior es suficiente para generar un código, pero los parámetros se pueden enlazar a propiedades de un bean, de modo que puedan variar en función de una lógica. Del mismo modo y como la generación se hace mediante ajax se puede recargar el componente de modo que se actualice a nuevos valores, esto se hará indicando el identificador del componente en la propiedad render del componente que desencadene la acción:

**Ejemplo.xhtml**

```
<a4j:commandLink render="codigodebarras"
    actionListener="#{barcodeSampleBean.generaCodigo}"
    styleClass="botonAplicacionTXT"
    value="#{bundle['barcode.generar']}">
    <h:graphicImage value="img/ico_nuevo.gif"
        alt="#{bundle['barcode.generar']}" />
</a4j:commandLink>
```

Los atributos del componente son los siguientes:

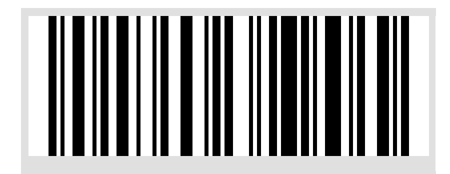
| Nombre atributo | Obligatorio | Descripción |
|---|---|---|
| id | SI | Identificador del componente, debe ser único en la página. |
| data | SI | Datos a codificar. |
| type | SI | Tipo de código a generar, los posibles valores son: AZTEC, CODABAR, CODE_128, CODE_39, DATA_MATRIX, EAN_13, EAN_8, ITF, PDF_417, QR_CODE, UPC_A. |
| rendered | NO | Valor true/false para seleccionar si este componente se va a mostrar. Por defecto es true. |
| generateCode | NO | Valor true/false para seleccionar si se debe invocar el servicio de generación del código, se puede utilizar esta variable para controlar previamente que la generación va a ser correcta y si no lo va a ser no generar el código. |
| width | NO | Anchura en pixels de la imagen generada, parámetro válido en los tipos CODABAR, CODE_128, CODE_39, EAN_13, EAN_8, ITF, QR_CODE, UPC_A. |
| height | NO | Altura en pixels de la imagen generada, parámetro válido en los tipos CODABAR, CODE_128, CODE_39, EAN_13, EAN_8, ITF, QR_CODE, UPC_A. |
| margin | NO | Margen que se deja en la imagen generada, parámetro válido en los tipos: CODABAR, CODE_128, CODE_39, QR_CODE. |
| characterSet | NO | Character Set para el tipo de código Aztec. |
| errorCorrection | NO | Grado de corrección de error para los códigos QR y Aztec. |
| dataMatrixShape | NO | Parámetro para configurar el tipo de código Data Matrix, sus posibles valores son: FORCE_NONE, FORCE_RECTANGLE, FORCE_SQUARE. |
| maxSize | NO | Tamaño máximo para el código Data Matrix, debe especificarse como un par de números separados por un espacio que representan ancho y alto. |
| minSize | NO | Tamaño mínimo para el código Data Matrix, debe especificarse como un par de números separados por un espacio que representan ancho y alto. |
| pdf417Compact | NO | Valor TRUE/FALSE para especificar si se debe usar el modo compacto en la generación del código de tipo PDF417. Su valor por defecto es FALSE. |
| pdf417Compaction | NO | Especifica el tipo de generación para el código PDF417, los posibles valores son: AUTO, BYTE, NUMERIC, TEXT. |

| | | |
|---|---|---|
| pdf417Dimensions | NO | Especifica el mínimo y máximo número de filas y columnas en la generación del código PDF417, debe especificarse como 4 números separados por espacios que representan el mínimo número de columnas, máximo número de columnas, mínimos número de filas, máximo número de filas. |
| styleClass | NO | Clase css del panel que engloba el componente. |
| onclick | NO | Código javascript que se ejecutará cuando se haga click sobre el código generado. |

### 4.3. Principales tipos de códigos

A continuación de muestra un ejemplo de generación de los principales tipos de código, codificando el texto 1234567890:

- CODE_128, tamaño 200 x 70, el código se corresponde con el ejemplo mostrado anteriormente



- QR_CODE, tamaños 120 x 120

**Ejemplo_QR.xhtml**

```
<atlas:barcode id="codigodebarras" barcodeType="QR_CODE"
            data="1234567890" height="120" width="120" />
```



- PDF417

**Ejemplo_PDF417.xhtml**

```
<atlas:barcode id="codigodebarras" barcodeType="PDF_417"
          data="1234567890"/>
```



En el ejemplo de la aplicación de componentes se pueden probar el resto de códigos, así que como los anteriores modificando cualquier parámetro.

#### 4.4. Servicio de código de barras

Adicionalmente al componente visual, se ha creado un servicio para generar/validar la generación de códigos, este servicio es el que utiliza internamente el componente, pero es posible utilizarlo al margen de él. Para utilizarlo se debe crear en el bean que los utilice una referencia a la clase atlas.componentes.facade.AtlasFacade, definida en el contexto de Spring con el id “atlasFacade”.

Esta clase tiene dos métodos:

- *public Barcode createBarcode(BarcodeParam param) throws AtlasBarcodeException*
- *public void validaParamBarcode(BarcodeParam param) throws AtlasBarcodeException*

El primer método devuelve una instancia de la clase *atlas.componentes.service.barcode.Barcode* configurada para generar el código, esta clase contiene métodos para generar el código a un fichero, a un *OutputStream* o generando una instancia de *BufferedImage* para un tratamiento posterior.

El segundo método se encarga de validar los parámetros, lanzando una excepción si alguno de los parámetros es incorrecto, los errores no sólo pueden ser por falta de parámetros, sino que pueden depender del tipo de código, ya que algunos códigos exigen un formato especial para los datos, por ejemplo, los códigos de tipo Aztec deben empezar por “A”, “B”, “C” o “D” y terminar por “T”, “N”, “*” o “E”.

La clase que se utiliza en para el paso de parámetros es *atlas.componentes.service.barcode.BarcodeParam*, y contiene los mismos parámetros especificados en la lista de atributos del componente, con las siguientes

salvedades:

- El tipo de código se especifica utilizando el tipo enumerado *atlas.componentes.service.barcode.BarcodeType*
- El parámetro dataMatrixShape se especifica mediante el tipo enumerado *com.google.zxing.datamatrix.encoder.SymbolShapeHint*
- Los parámetros maxSize y minSize se especifican mediante la clase *com.google.zxing.Dimension*
- El parámetro pdf417Compaction se especifica mediante el tipo enumerado *com.google.zxing.pdf417.encoder.Compaction*
- El parámetro pdf417Dimensions se especifica mediante la clase *com.google.zxing.pdf417.encoder.Dimensions*

### 4.5. Recomendaciones y buenas prácticas

Se recomienda no aumentar demasiado las cantidades de anchura y altura para la buena visualización del componente.

### 4.6. Ejemplo de Uso

Se puede ver un ejemplo de dicho componente en la aplicación de demostración de componentes, bajo la siguiente secuencia de navegación:

Inicio > Otros componentes Atlas > Código de barras

## 5. PREGUNTAS MÁS FRECUENTES

En este apartado se incluyen una lista de preguntas más frecuentes sobre el componente.

**Pregunta**: ¿Dónde puedo encontrar información general sobre los componentes?
**Respuesta**: En la aplicación de demostración de los componentes del Framework Atlas

**Pregunta**: ¿Cómo puedo averiguar todas las funcionalidades disponibles en la librería Zxing?
**Respuesta**: Accediendo a la página Web de la librería que se indica en la sección de *Enlaces Relacionados*.

**Pregunta**: ¿Cómo se ha implementado el componente en su capa de presentación?
**Respuesta**: El componente se compone de unos servicio que generan los diferentes códigos y de una componente facelets que hace uso de los servicios para mostrarlo en una aplicación web.

**Pregunta**: ¿Cómo puedo validar los parámetros en la generación de un código?
**Respuesta**: En el apartado 4.4 se explica el uso de los servicios, uno de los métodos permite validar los parámetros de entrada de modo que podamos saber si esos parámetros van a generar o no un código correcto. Utilizando ese método de validación combinado con la propiedad *generateCode* se puede controlar que el componente sólo se genere en el caso en que los parámetros sean correctos. El ejemplo de la aplicación de componente valida los parámetros de este modo.

## 6. ENLACES RELACIONADOS

| Producto | URL |
|---|---|
| Ajax4JSF | http://www.jboss.org/jbossrichfaces/ |
| Zxing | http://code.google.com/p/zxing/ |
| Commons BeanUtils | commons.apache.org/beanutils/ |
| Commons Configurations | http://commons.apache.org/configuration/ |
| Facelets | https://facelets.dev.java.net/ |
| Hibernate | http://www.hibernate.org/ |
| Hibernate Annotations | http://www.hibernate.org/hib_docs/annotations/reference/en/html_single/ |
| JAXB | http://java.sun.com/webservices/jaxb/ |
| JPA | http://java.sun.com/developer/technicalArticles/J2EE/jpa/ |
| JSF | http://java.sun.com/javaee/javaserverfaces/ |
| JSFUnit | http://www.jboss.org/jsfunit/ |
| Log4J | http://logging.apache.org/log4j/ |
| MyFaces Core | http://myfaces.apache.org/ |
| RichFaces | http://www.jboss.org/jbossrichfaces/ |
| Spring | http://www.springframework.org/ |
| Spring Security | http://www.springframework.org/ |
| Tomahawk | http://myfaces.apache.org/tomahawk/ |
| Velocity | http://velocity.apache.org/ |